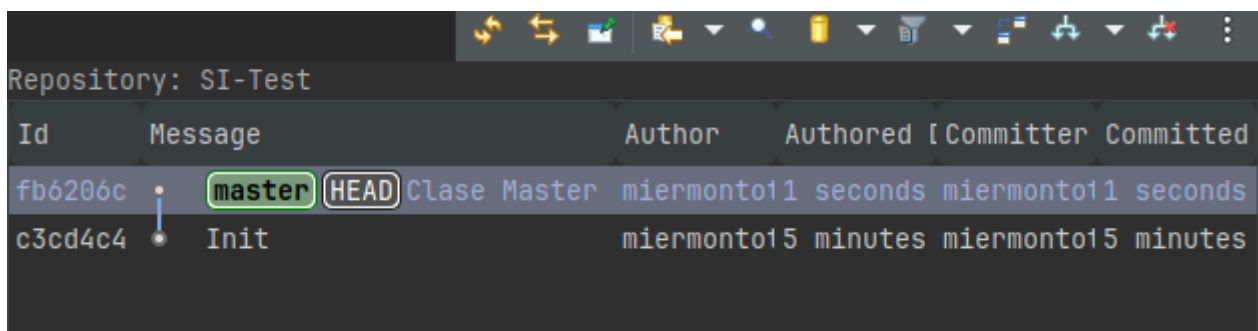


Práctica para ejercitar el uso de Git

Juan Francisco Mier Montoto, UO283319
Sistemas de Información, EPI Gijón 22-23

1. Realizar un **merge** de una rama de trabajo en **master** (sin conflictos) cuando **master** ha sido modificado después de hacer el branch

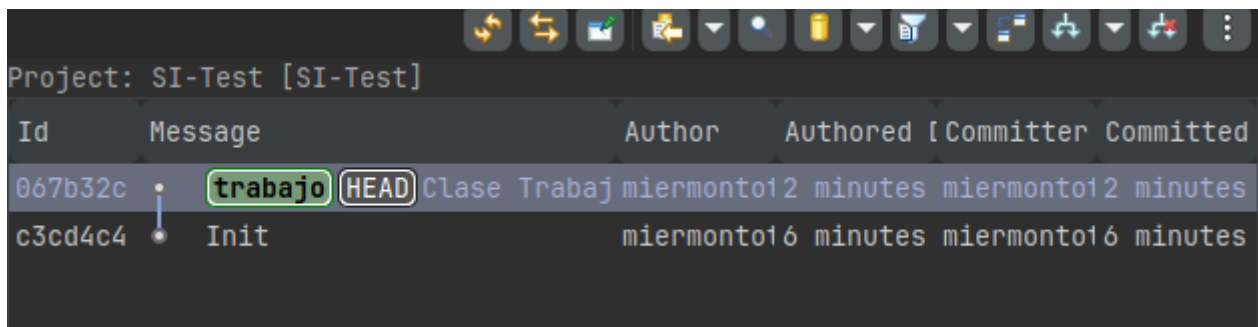
Estado de la rama **master** previo a los merges



Id	Message	Author	Authored	[Committer	Committed
fb6206c	master HEAD Clase Master	miermonto	11 seconds	miermonto	11 seconds
c3cd4c4	Init	miermonto	15 minutes	miermonto	15 minutes

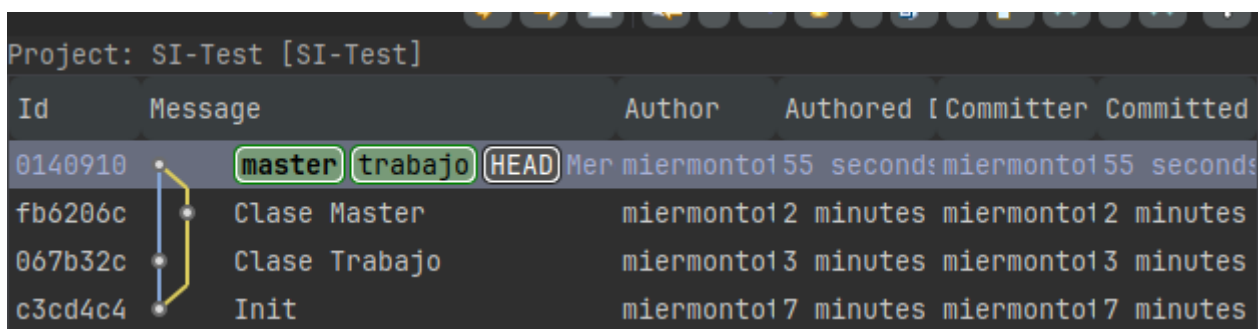
El commit **Clase Master** es posterior a la creación de la branch **trabajo**.

Estado de la rama **trabajo** previo a los merges



Id	Message	Author	Authored	[Committer	Committed
067b32c	trabajo HEAD Clase Trabajo	miermonto	12 minutes	miermonto	12 minutes
c3cd4c4	Init	miermonto	16 minutes	miermonto	16 minutes

Merge de la rama **trabajo** en **master**

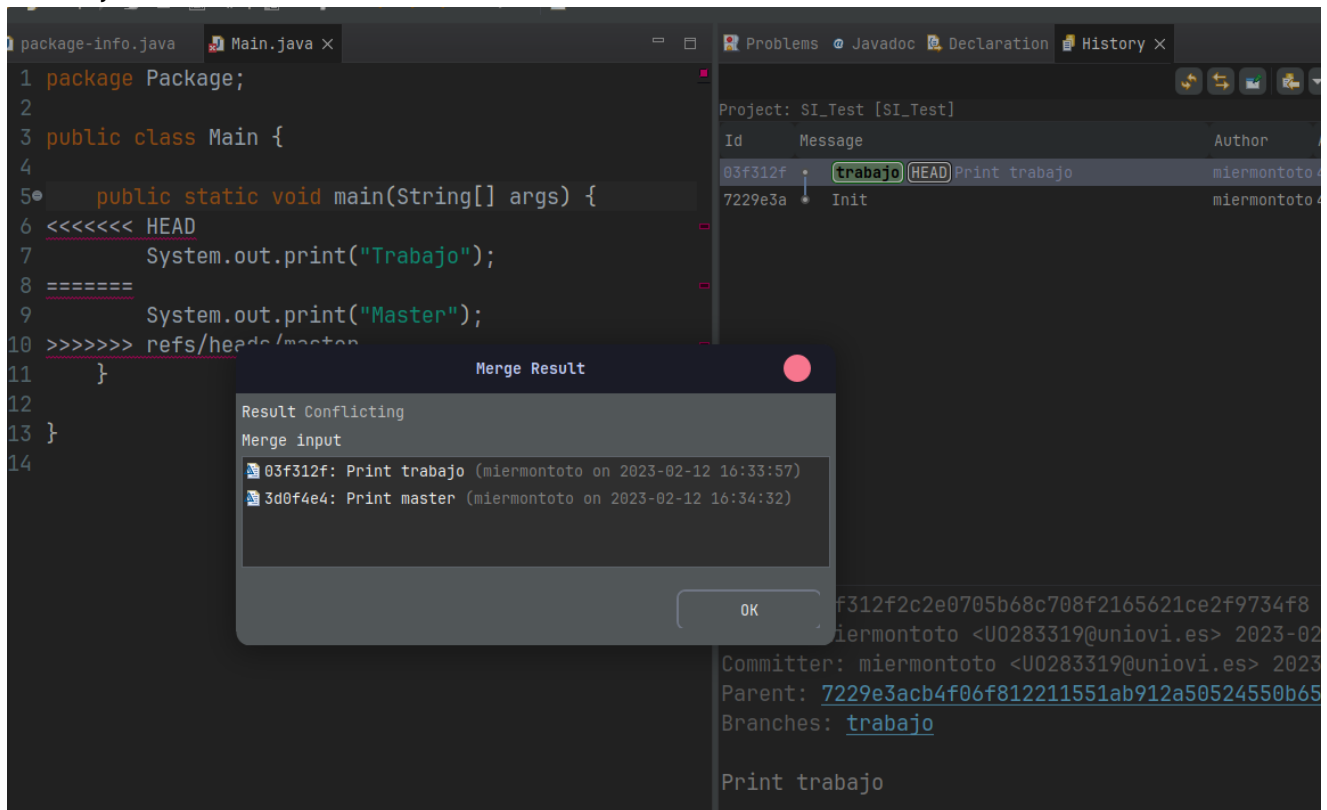


Id	Message	Author	Authored	[Committer	Committed
0140910	master trabajo HEAD Mer	miermonto	155 seconds	miermonto	155 seconds
fb6206c	Clase Master	miermonto	12 minutes	miermonto	12 minutes
067b32c	Clase Trabajo	miermonto	13 minutes	miermonto	13 minutes
c3cd4c4	Init	miermonto	17 minutes	miermonto	17 minutes

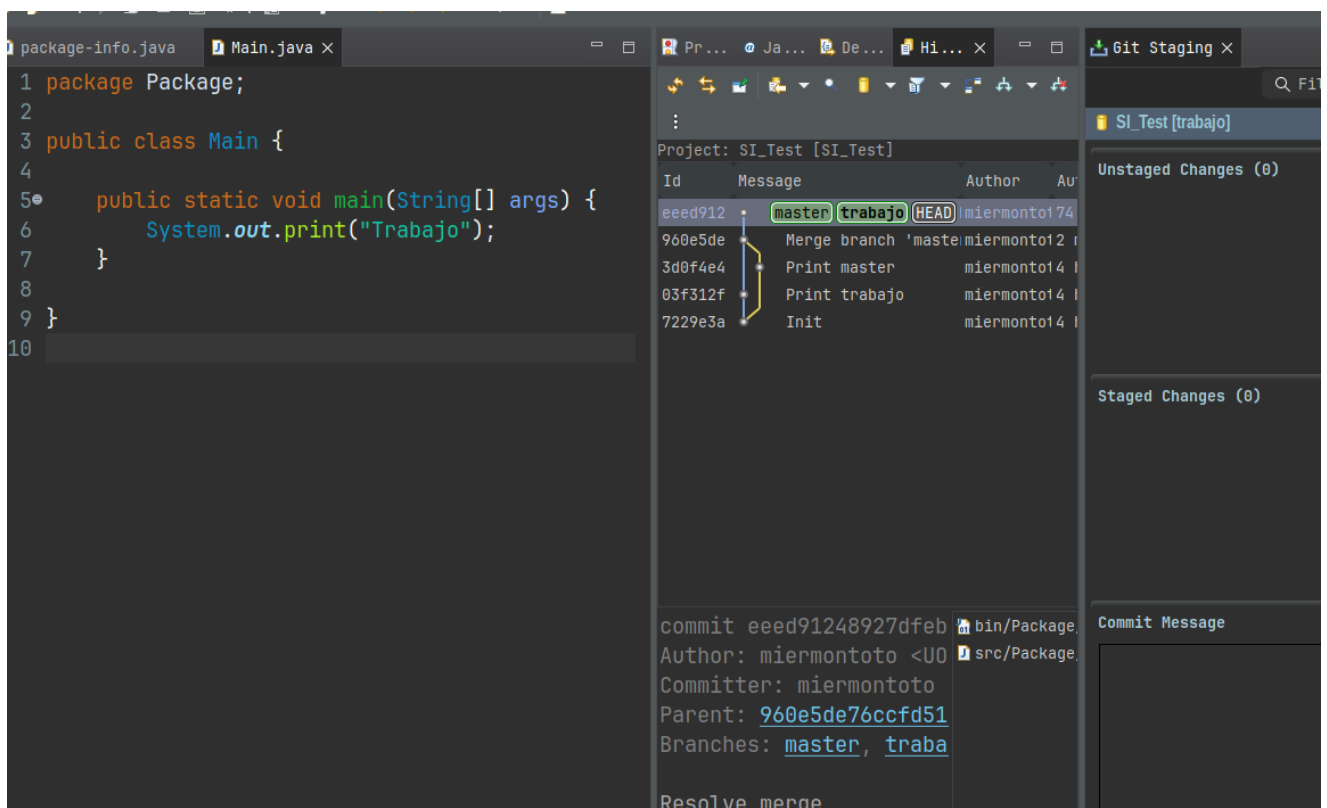
Al hacer este merge, se incorporan los cambios en la rama **trabajo** en **master**. Puesto que no hay conflictos, el merge hace **fast-forward** y se realiza de manera limpia y sin perder información.

2. Realizar un **merge** de **master** en una rama de trabajo, cuando **master** ha sido modificado después de hacer el branch y hay una línea que ha cambiado en ambas ramas (resolver conflicto)

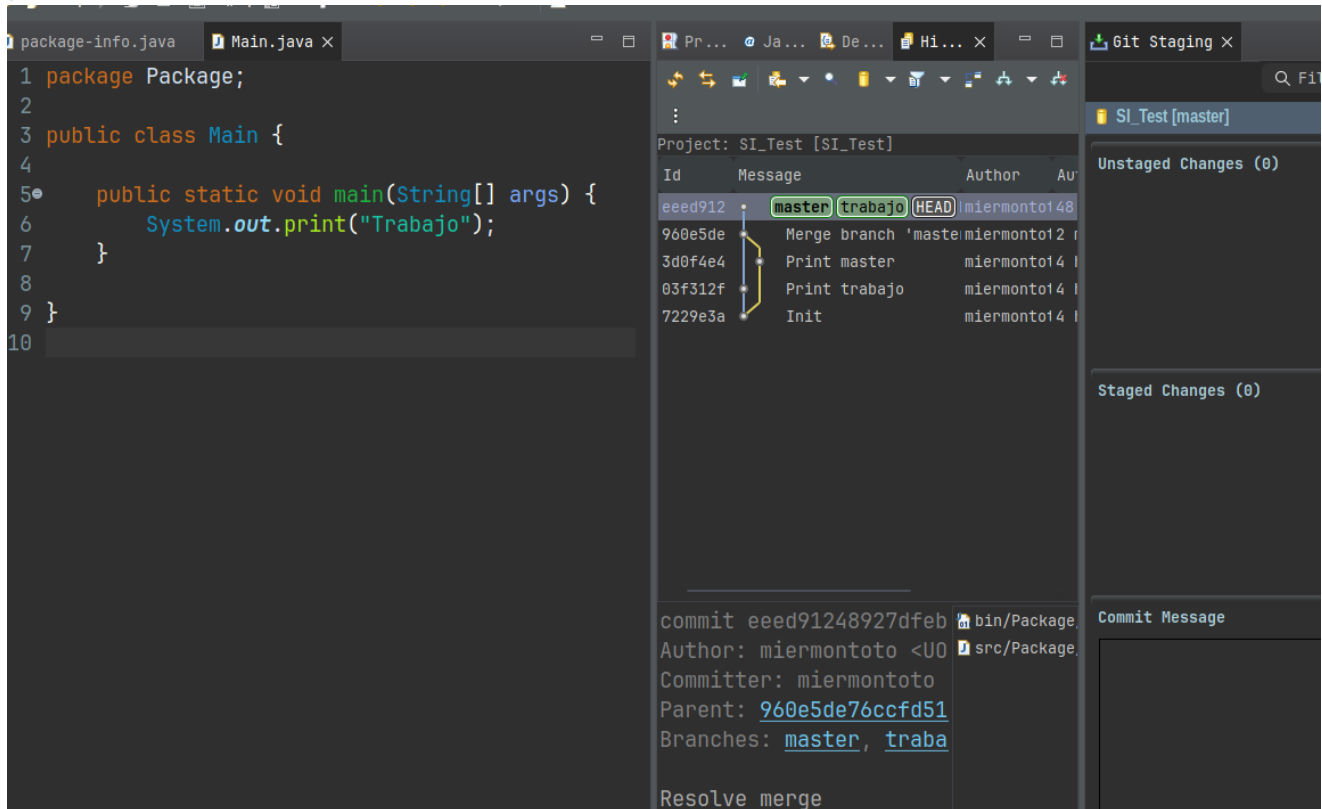
Mismo esquema que el ejercicio anterior. Al crear dos ramas y combinarlas, aparece este mensaje:



Se resuelven los conflictos:

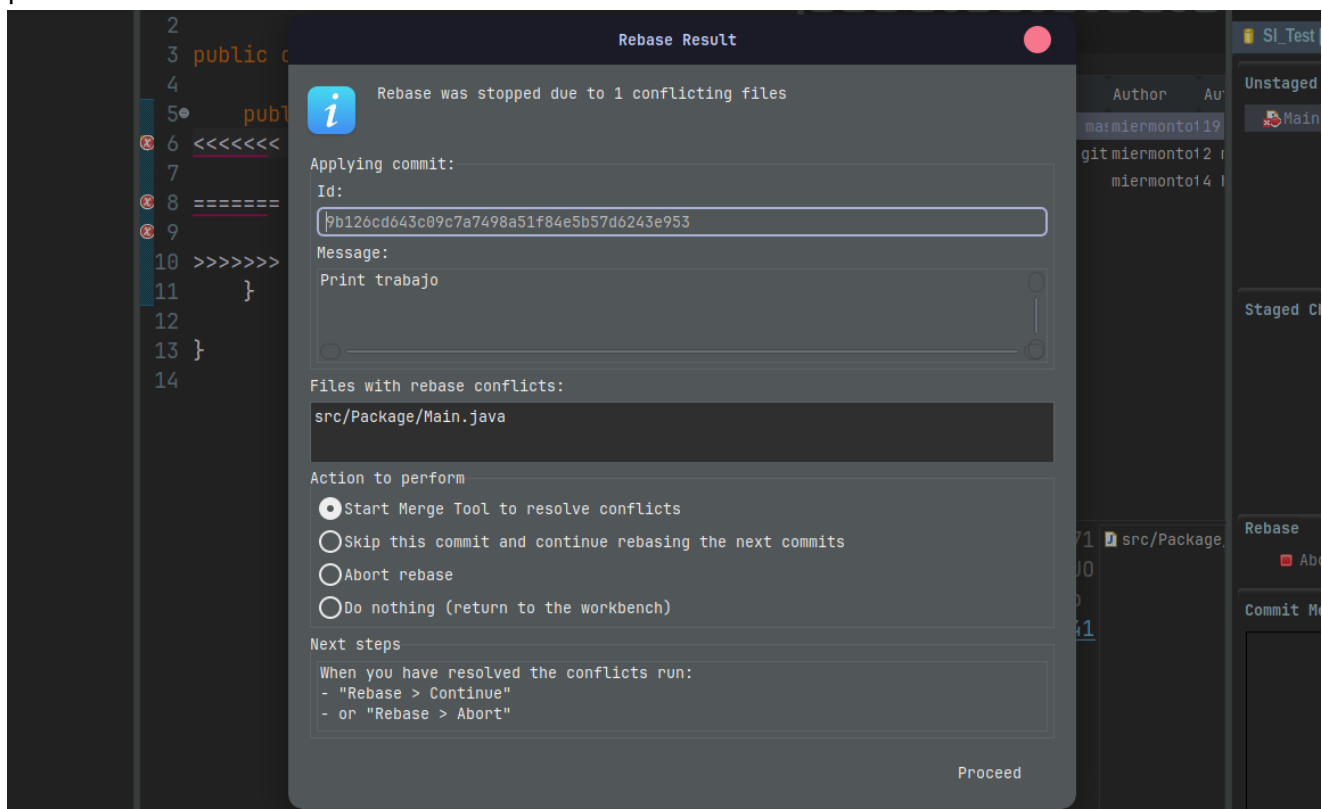


Ahora, tan solo hace falta hacer **merge** con la rama **master** igual que en el ejercicio anterior y todos los cambios estarán actualizados:







3. Utilizar **rebase** en vez de **merge** para lo anterior

En la misma situación que en el ejercicio anterior, al hacer **rebase** se obtiene la siguiente pantalla:



Tras arreglar los conflictos a través del **Merge Tool**, se continúa con el rebase y se finaliza:

Project: SI_Test [SI_Test]

Id	Message	Author	Authored [Committer	Committed	
6d7fde0	 trabajo (HEAD) Fix rebase	miermontot	4 minutes	miermontot	4 minutes
64cf2b2	 master Print master	miermontot	6 minutes	miermontot	6 minutes
2ba7ae0	 remove bin files, gitignore	miermontot	7 minutes	miermontot	7 minutes
7229e3a	 Init	miermontot	4 hours ago	miermontot	4 hours ago

En este caso, todos los cambios quedan rebasados en la rama **trabajo**. En caso de querer guardar todos los cambios en la rama **master**, se debería hacer rebase en el otro sentido. Puesto que es un rebase, no se aprecian las ramas de las que vienen los cambios y la vista resultante es sencilla.

4. Realizar varios commits en una rama de trabajo y usar rebase interactivo para agruparlos en uno solo

Sobre la misma estructura anterior, se realizan varios commits de prueba (en la rama **trabajo**):

```
package Package;

public class Main {

    public static void main(String[] args) {
        System.out.print("Print");
    }

    public void test() {
        return;
    }

    public int getFour() {
        return 4;
    }
}
```

File: SI_Test/src/Package/Main.java [SI_Test]

Id	Message	Author	Authored [Comm
c281e98	trabajo (HEAD) getFour	miermontot	12 seconds mien
c10a9f4	Test method	miermontot	122 seconds mien
7b3f4fe	Print	miermontot	147 seconds mien
7229e3a	Init	miermontot	15 hours ago mien

Al hacer el rebase interactivo, se hace **squash** sobre todos los commits excepto el primero:

Repository: SI_Test

Id	Message	Author	Authored [Comm
7b3f4fe	Print	miermontot	14 minutes mien
2ba7ae0	master (HEAD) remove b:miermontot	122 minutes mien	
7229e3a	Init	miermontot	15 hours ago mien

SI_Test|Interactive rebase

Start Continue Skip Abort Refresh Pick Skip Edit Squash

Status	Step	Action	Commit Id	Message	Aut
	1.	Pick	7b3f4fe	Print	mie
	2.	Squash	c10a9f4	Test method	mie
	3.	Squash	c281e98	getFour	mie

Al final, se colapsan todos los commits en el primero, en este caso en **Print**:

```
package Package;

public class Main {

    public static void main(String[] args) {
        System.out.print("Print");
    }

    public void test() {
        return;
    }

    public int getFour() {
        return 4;
    }
}
```

Repository: SI_Test

Id	Message	Author	Authored [Comm
165867b	trabajo (HEAD) Print T	miermonto	15 minutes mier
7b3f4fe	Print	miermonto	15 minutes mier
2ba7ae0	master remove bin f	miermonto	123 minutes: mier
7229e3a	Init	miermonto	15 hours a: mier